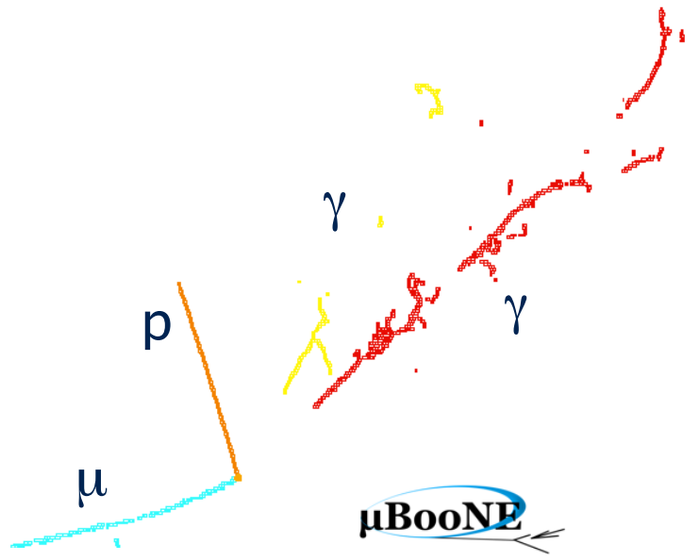


Pandora Exercise 1: Input from LArSoft

J. S. Marshall for the Pandora Team

MicroBooNE Pandora Workshop

July 11-14th 2016, Cambridge

Running Pandora In LArSoft

Pre-requisite: LArSoft v05_13_00 (inc. LArPandoraContent v02_07_04) or later

Run Pandora to (re-)process a MicroBooNE BNB Nu 'reco2' file.

Attempt to run Pandora internal event display within LArSoft

Persist input events in Pandora .pndr or .xml file formats for usage in later exercises.

```
These instructions have been tested with:

-Scientific Linux CERN SLC release 6.7, gcc 4.9.3, ROOT 5.34.32, using LArSoft v05_13_00 via CVMFS

Some instructions will doubtless require modification for different systems/setups
```

Run Pandora Blind

```
source /cvmfs/uboone.opensciencegrid.org/products/setup_uboone.sh
setup uboonecode v05_13_00 -q e9:prof

# List some available .fcl files
ls $UBOONECODE_DIR/job

cp $UBOONECODE_DIR/job/reco_uboone_mcc7_driver_stage2.fcl ./myreco_uboone_mcc7_driver_stage2.fcl
```

Informed choice, as required

```
#include "reco_uboone_mcc7_driver_common.fcl"

process_name: PandoraWorkshop

services.DetectorClocksService.InheritClockConfig:  false
services.TFileService.fileName: “reco_stage_2_hist.root"

physics.reco: [ @sequence::microboone_reco_mcc7_stage2 ]
physics.trigger_paths: [ reco ]
outputs.out1.fileName: "%ifb_%tc_reco2.root"
outputs.out1.dataTier: "reconstructed"
source.inputCommands: ["keep *_*_*_*", "drop *_*_*_McRecoStage2" ]
```

Will likely need to change process name, if reprocessing a reco2 file

Run Pandora Blind

```
lar -c myreco_uboone_mcc7_driver_stage2.fcl -n 5 /path/to/reco2/file.root
```

Typical starting point: /pnfs/uboone/scratch/users/uboonepro/mcc7/v05_08_00/reco2/prodgenie_bnb_nu_uboone

OR /r05/dune/mcproduction_v05_08_00/larsoft_output_reco2* on Cambridge HEP systems

```
-bash-4.1$ lar -c myreco_uboone_mcc7_driver_stage2.fcl -n 2 /r05/dune/mcproduction_v05_08_00/larsoft_output_reco2_bnb_nu/
prodgenie_bnb_nu_uboone_0_20160409T043612_gen_36bd0b2a-31c1-4f85-84c7-
ebc23d7484ed_20160409T070008_g4_20160413T141922_detsim_20160418T045659_reco1_20160502T155456_reco2.root
Using channel statuses from conditions database
Using pedestals from conditions database
%MSG-w OpDigiProperties:  lar 10-Jun-2016 15:20:48 BST JobSetup
    OpDigiProperties using analytical function for WF generation.
%MSG
%MSG-w OpDigiProperties:  lar 10-Jun-2016 15:20:48 BST JobSetup
    Generating gain for each pmt.
        High gain mean: 20 ADC/p.e.
        Low  gain mean: 2 ADC/p.e.
        PMT-to-PMT gain spread : 0.05
        Intrinsic gain spread  : 0.05

%MSG
2 TPC ASIC configs are activated
224 channels read in
Config map first/last channels: 2016 2383
CalibResponseTOffsets: 0 0 0
Current E field = 0.273 KV/cm, Ratio of drift velocities = 1.44085, timeScaleFactor = 1.35533
10-Jun-2016 15:20:48 BST  Initiating request to open file /r05/dune/mcproduction_v05_08_00/larsoft_output_reco2_bnb_nu/
prodgenie_bnb_nu_uboone_0_20160409T043612_gen_36bd0b2a-31c1-4f85-84c7-
ebc23d7484ed_20160409T070008_g4_20160413T141922_detsim_20160418T045659_reco1_20160502T155456_reco2.root
10-Jun-2016 15:20:48 BST  Successfully opened file /r05/dune/mcproduction_v05_08_00/larsoft_output_reco2_bnb_nu/
prodgenie_bnb_nu_uboone_0_20160409T043612_gen_36bd0b2a-31c1-4f85-84c7-
ebc23d7484ed_20160409T070008_g4_20160413T141922_detsim_20160418T045659_reco1_20160502T155456_reco2.root
%MSG-w FastCloning:  PostOpenFile 10-Jun-2016 15:20:49 BST BeforeEvents
Fast cloning deactivated for this input file due to information in FileBlock.
%MSG
...
```

Some samples also available here (docdb username/password)

Don't 'see' what's happening, but you do get the output products in root file (discuss later)

Enable Visualisation

Pandora Visualisation uses ROOT TEVE. This should work nicely on most linux and Mac systems, but can sometimes be frustrating, esp. when embedded in a complex application, such as LArSoft.

```
cp $LARPANDORA_DIR/scripts/PandoraSettings_MicroBooNE_Neutrino.xml
   ./MyPandoraSettings_MicroBooNE_Neutrino.xml

# Edit .fcl file and PandoraSettings file as shown below
```

```
#include "reco_uboone_mcc7_driver_common.fcl"

process_name: PandoraWorkshop

services.RootGraphicsEnablingService: {}

services.DetectorClocksService.InheritClockConfig:  false
services.TFileService.fileName: "reco_stage_2_hist.root"

physics.producers.pandoraNu.HitFinderModuleLabel:   "gaushit"
physics.producers.pandoraNu.ConfigFile:             "MyPandoraSettings_MicroBooNE_Neutrino.xml"

physics.reco: [ pandoraNu ]
physics.trigger_paths: [ reco ]
outputs.out1.fileName: "%ifb_%tc_reco2.root"
outputs.out1.dataTier: "reconstructed"
source.inputCommands: ["keep *_*_*_*", "drop *_*_*_McRecoStage2" ]
```

Addresses unfortunate 'race' for resources in ROOT (gets there first!)

Just use all input Hits and point to local PandoraSettings file

Just run pandoraNu

Enable Visualisation

Pandora Visualisation uses ROOT TEVE. This should work nicely on most linux and Mac systems, but can sometimes be frustrating, esp. when embedded in a complex application, such as LArSoft.

```
cp $LARPANDORA_DIR/scripts/PandoraSettings_MicroBooNE_Neutrino.xml
   ./MyPandoraSettings_MicroBooNE_Neutrino.xml

# Edit .fcl file and PandoraSettings file as shown below
```

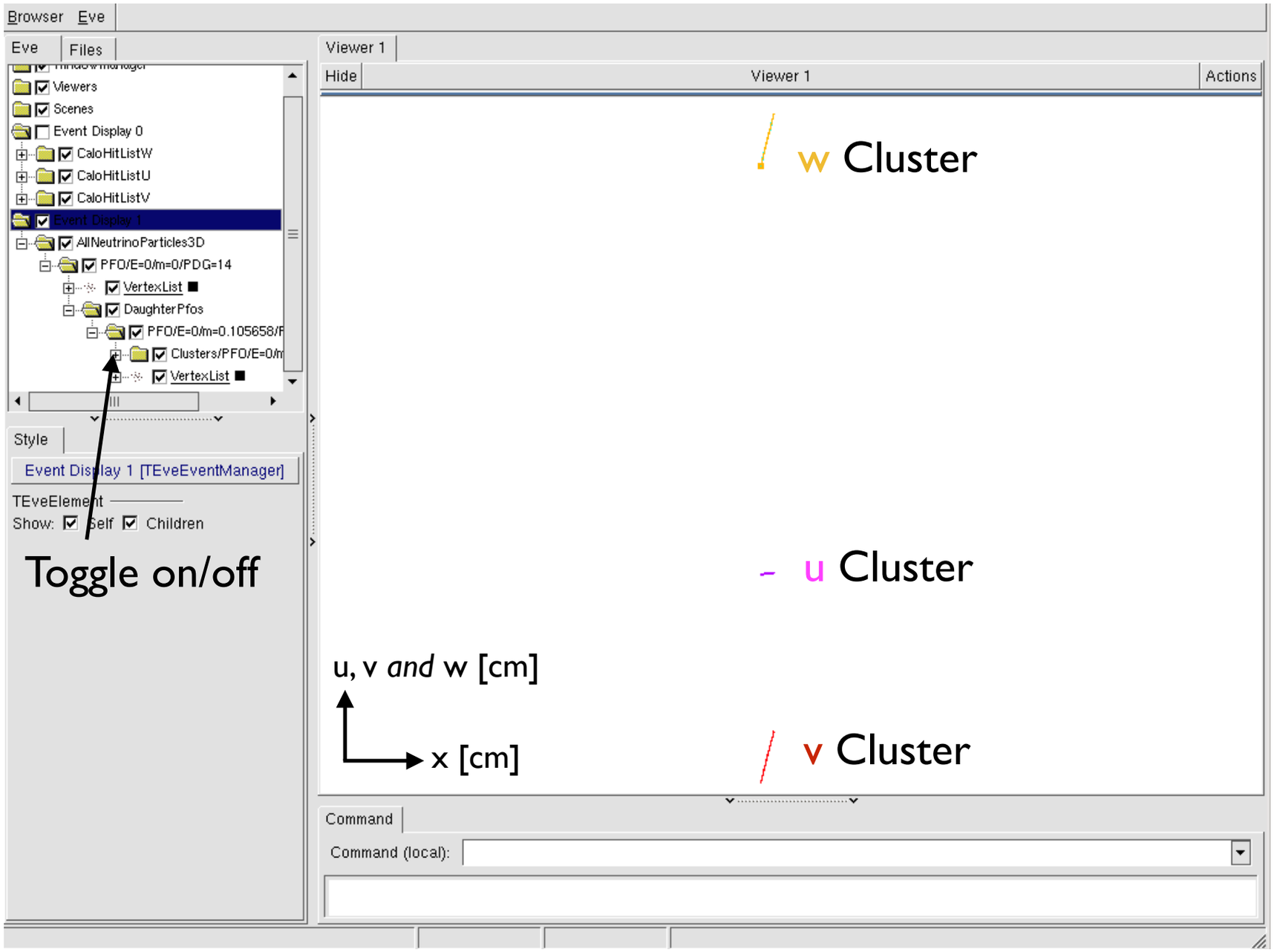
```xml
<!-- Pandora settings xml file -->
<pandora>
    <!-- GLOBAL SETTINGS -->
    <IsMonitoringEnabled>true</IsMonitoringEnabled>
    <ShouldDisplayAlgorithmInfo>true</ShouldDisplayAlgorithmInfo>
    <SingleHitTypeClusteringMode>true</SingleHitTypeClusteringMode>

    <!-- PLUGIN SETTINGS -->
    <MuonPlugin>LArMuonId</MuonPlugin>

    <!-- NEUTRINO-INDUCED EVENT RECONSTRUCTION -->
    <algorithm type = "LArListPreparation">
        <OnlyAvailableCaloHits>true</OnlyAvailableCaloHits>
        <OutputCaloHitListNameW>CaloHitListW</OutputCaloHitListNameW>
        <OutputCaloHitListNameU>CaloHitListU</OutputCaloHitListNameU>
        <OutputCaloHitListNameV>CaloHitListV</OutputCaloHitListNameV>
        <FilteredCaloHitListName>CaloHitList2D</FilteredCaloHitListName>
        <CurrentCaloHitListReplacement>CaloHitList2D</CurrentCaloHitListReplacement>
        <OutputMCParticleListNameU>MCParticleListU</OutputMCParticleListNameU>
        <OutputMCParticleListNameV>MCParticleListV</OutputMCParticleListNameV>
        <OutputMCParticleListNameW>MCParticleListW</OutputMCParticleListNameW>
        <OutputMCParticleListName3D>MCParticleList3D</OutputMCParticleListName3D>
        <CurrentMCParticleListReplacement>MCParticleList3D</CurrentMCParticleListReplacement>
        <MipEquivalentCut>0.</MipEquivalentCut>
    </algorithm>

    <algorithm type = "LArVisualMonitoring">
        <CaloHitListNames>CaloHitListW CaloHitListU CaloHitListV</CaloHitListNames>
    </algorithm>
    ...
```

Turn-on PandoraMonitoring support and (might as well) display Pandora algorithm names and ids.

Note LArVisualMonitoring algorithm instance (much more on this soon)

Enable Visualisation

```
lar -c myreco_uboone_mcc7_driver_stage2.fcl -n 5 /path/to/reco2/file.root
```



[If all is well with ROOT!]

```
Press return to continue ...
```

Write Events in Pandora Format

- **Pandora persistency allows input objects (Hits, MCParticles, Gaps etc.) to be serialised in .pndr files (small, portability not guaranteed) or .xml files (large, but compressible).**
  - No longer need full client/translation app to develop or test algs: can move to lightweight environment where Entry Point constructs Pandora instance and runs reconstruction.
  - Enables development without delays or complications introduced by parent software framework and build system: rebuild and run in seconds, making for healthy development.
  - Simple Makefile option and command line app: in realm of standard, well documented C++

```
// ATTN: Edited for slide display; inc. removal of API return value checks
int main(int argc, char *argv[])
{
    Parameters parameters;

    if (!parameters.ParseCommandLine(argc, argv))
        return 1;

    const pandora::Pandora *const pPandora(new pandora::Pandora());
    LArContent::RegisterAlgorithms(*pPandora);
    PandoraApi::ReadSettings(*pPandora, parameters.m_pandoraSettingsFile);

    unsigned int nEvents(0);
    while (nEvents++ < parameters.m_nEventsToProcess)
    {
        PandoraApi::ProcessEvent(*pPandora);
        PandoraApi::Reset(*pPandora);
    }

    delete pPandora;
    return 0;
}
```

- Self-describing input objects; algs don't need to worry how/where object properties were calculated.
- Objects serialised/deserialised by Pandora, following requests from EventReading, EventWriting algs.

```
<!-- ALGORITHM SETTINGS -->
<algorithm type = "LArEventReading">
     <EventFileName>/PATH/TO/Events.pndr</EventFileName>
     <ShouldReadEvents>true</ShouldReadEvents>
     <SkipToEvent>0</SkipToEvent>
</algorithm>
```

Write Events in Pandora Format

```
cp $LARPANDORA_DIR/scripts/PandoraSettings_Write.xml ./MyPandoraSettings_Write.xml

# Edit .fcl file and PandoraSettings_Write file as shown below
```

Add a pandoraWriter producer (identical to pandoraNu, except MCParticles enabled and different PandoraSettings file)

```
#include "reco_uboone_mcc7_driver_common.fcl"

process_name: PandoraWorkshop

services.RootGraphicsEnablingService: {}

services.DetectorClocksService.InheritClockConfig:  false
services.TFileService.fileName: "reco_stage_2_hist.root"

# Only difference microboone_pandora and microboone_pandorawriter is that writer enables mc particle production
physics.producers.pandoraWriter:                        @local::microboone_pandorawriter
physics.producers.pandoraWriter.HitFinderModuleLabel:   "gaushit"
physics.producers.pandoraWriter.ConfigFile:             "MyPandoraSettings_Write.xml"

physics.producers.pandoraNu.HitFinderModuleLabel:       "gaushit"
physics.producers.pandoraNu.ConfigFile:                 "MyPandoraSettings_MicroBooNE_Neutrino.xml"

physics.reco: [ pandoraNu, pandoraWriter ]
physics.trigger_paths: [ reco ]
outputs.out1.fileName: "%ifb_%tc_reco2.root"
outputs.out1.dataTier: "reconstructed"
source.inputCommands: ["keep *_*_*_*", "drop *_*_*_McRecoStage2" ]
```

Just run pandoraNu and pandoraWriter; turn-off pandoraNu if you like (instances independent)

Write Events in Pandora Format

```
cp $LARPANDORA_DIR/scripts/PandoraSettings_Write.xml ./MyPandoraSettings_Write.xml

# Edit .fcl file and PandoraSettings_Write file as shown below
```

```xml
<!-- Pandora settings xml file -->

<pandora>
    <!-- GLOBAL SETTINGS -->
    <IsMonitoringEnabled>false</IsMonitoringEnabled>
    <ShouldDisplayAlgorithmInfo>false</ShouldDisplayAlgorithmInfo>
    <SingleHitTypeClusteringMode>true</SingleHitTypeClusteringMode>

    <!-- PLUGIN SETTINGS -->
    <MuonPlugin>LArMuonId</MuonPlugin>

    <!-- ALGORITHM SETTINGS -->
    <!--algorithm type = "LArEventReading">
        <EventFileName>INPUT_XML_OR_PNDR_FILE</EventFileName>
        <ShouldReadEvents>true</ShouldReadEvents>
        <SkipToEvent>0</SkipToEvent>
    </algorithm-->

     <algorithm type = "LArEventWriting">
        <EventFileName>MyPandoraEvents.xml</EventFileName>
        <ShouldWriteEvents>true</ShouldWriteEvents>
        <ShouldOverwriteEventFile>true</ShouldOverwriteEventFile>
        <ShouldWriteMCRelationships>true</ShouldWriteMCRelationships>
        <ShouldWriteTrackRelationships>true</ShouldWriteTrackRelationships>
    </algorithm>
</pandora>
```

LArEventWriting alg instance - specify output file name and format (.pndr or .xml)

Write Events in Pandora Format

```
lar -c myreco_uboone_mcc7_driver_stage2.fcl -n 5 /path/to/reco2/file.root
```

E.g. `MyPandoraEvents.xml` snippet


```xml
<Event>
    <CaloHit>
        <CellGeometry>0</CellGeometry>
        <PositionVector>250.787 0 1036.75</PositionVector>
        <ExpectedDirection>0 0 1</ExpectedDirection>
        <CellNormalVector>0 0 1</CellNormalVector>
        <CellThickness>0.3</CellThickness>
        <NCellRadiationLengths>0.0357143</NCellRadiationLengths>
        <NCellInteractionLengths>0.00595238</NCellInteractionLengths>
        <Time>0</Time>
        <InputEnergy>63.0018</InputEnergy>
        <MipEquivalentEnergy>0.87453</MipEquivalentEnergy>
        <ElectromagneticEnergy>0.000306085</ElectromagneticEnergy>
        <HadronicEnergy>0.000306085</HadronicEnergy>
        <IsDigital>0</IsDigital>
        <HitType>6</HitType>
        <HitRegion>2</HitRegion>
        <Layer>0</Layer>
        <IsInOuterSamplingLayer>0</IsInOuterSamplingLayer>
        <ParentCaloHitAddress>389</ParentCaloHitAddress>
        <CellSize0>0.5</CellSize0>
        <CellSize1>0.326799</CellSize1>
    </CaloHit>
    <MCParticle>
        <NuanceCode>0</NuanceCode>
        <Energy>0.00054429</Energy>
        <Momentum>-0.000183322 -2.22094e-05 3.21137e-05</Momentum>
        <Vertex>248.313 -101.187 941.107</Vertex>
        <Endpoint>248.313 -101.187 941.107</Endpoint>
        <ParticleId>11</ParticleId>
        <MCParticleType>3</MCParticleType>
        <Uid>394</Uid>
    </MCParticle>
    ...
```


Use as input to Pandora standalone development environment in later exercises

Next Exercise: Setup the Pandora Standalone Development Environment and add a new Algorithm